

(7) Japanese Patent Application Laid-Open No. JP1-155435 (1989)

“Square-Root Computation Method”

The following is an extract relevant to the present application.

5

The present invention relates to a square-root computation method.

The present invention performs the following computing equation for obtaining a square root from an operand with a radix of 2 by a square-root extraction method.

$$R_n = R_{n-1} - (4 * A_{n-1} + 1)$$

10

wherein  $R_{n-1}$  = residual in the previous cycle

$A_{n-1}$  = intermediate result up to the previous cycle

When the computation result is  $R_n < 0$ , “0” is provided as a partial solution while in other cases “1” is provided as a partial solution. Further, when  $R_n < 0$ ,  $R_{n-1}$  is provided as a partial residue while in other cases what is left-shifted from  $R_n$  by two bits is provided as a partial residue.

15

The partial solution as aforementioned is repeatedly added to a result which is left-shifted from the intermediate result up to the previous cycle by one bit, thereby obtaining a square root.

⑬ 日本国特許庁(JP)

⑭ 特許出願公開

⑮ 公開特許公報(A)

平1-155435

⑯ Int. Cl.<sup>4</sup>

G 06 F 7/552  
7/52

識別記号

庁内整理番号

7056-5B  
A-7056-5B

⑰ 公開 平成1年(1989)6月19日

審査請求 未請求 発明の数 1 (全10頁)

⑱ 発明の名称 平方根演算処理方式

⑲ 特 願 昭62-314477

⑳ 出 願 昭62(1987)12月11日

㉑ 発 明 者 柳 田 昌 宏 神奈川県川崎市中原区上小田中1015番地 富士通株式会社  
内

㉒ 発 明 者 松 崎 重 治 神奈川県川崎市中原区上小田中1015番地 富士通株式会社  
内

㉓ 出 願 人 富士通株式会社 神奈川県川崎市中原区上小田中1015番地

㉔ 代 理 人 弁理士 井 析 貞一

## 明 細 書

### 1. 発明の名称

平方根演算処理方式

### 2. 特許請求の範囲

基数2のオペランドから開平法によって平方根を求める演算式

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

但し、 $R_{n-1}$  : 前サイクルでの余り

$A_{n-1}$  : 前サイクル迄の中間結果

を計算し、 $R_n < 0$  であれば '0' を、それ以外であれば '1' を部分剰余とし、且つ、 $R_n < 0$  の場合には  $R_{n-1}$  を、それ以外の場合は  $R_n$  を2ビット左シフトしたものを部分剰余とし、

前サイクル迄の中間結果を1ビット左シフトした結果に、上記部分剰余を加えることを繰り返して平方根を求めるのに、

少なくとも、除数レジスタ(1)と、除倍数回路(2)と、該除倍数の結果を保持する減数レジスタ(3)とを備えた高基数非回復型除算装置に、オペ

ランドを置数する被開平数レジスタ(6)と、

上記置数されたオペランドを1演算サイクル毎に2ビット左シフトして、上記高基数非回復型除算装置の部分剰余レジスタ(5)に伝達する手段

(①)と、

上記演算サイクル毎に、上記除数レジスタ(1)を1ビット左シフトして帰還する手段(②)と、

上記部分剰余レジスタ(5)の値から上記除数レジスタ(1)の値を1ビット左シフトして4倍したものを減算したときに、加算器(4)から得られるキャリ信号を、上記除数レジスタ(1)の最下位ビットに帰還して、平方根の部分剰余を得る手段(③)と、

上記演算サイクル毎に、部分剰余レジスタ(5)を2ビット左シフト(④)したものの( $R_{n-1}$ )と、上記減算処理によって得られる部分剰余( $R_n$ )との何れかを、上記キャリ信号の値によって選択して、上記部分剰余レジスタ(5)に置数し、当該演算サイクルでの部分剰余とする手段(⑤)とを設けて、当該高基数非回復型除算装置を用いて平方根演

算処理を行う方式において、

前サイクルでの部分剰余を置数した部分剰余レジスタ(5) から減算すべき前サイクルの部分中間結果( $A_{n-1}$ )の $(A_{n-1}) \times 4$ を生成するのに、

1 サイクル前の中間結果(1) を1ビット左シフトして、'×4'を行う第1の手段と、

今回の演算サイクルで得られた部分解(キャリ)を1ビット左シフトして'×4'を行うと同時に、該部分解(キャリ)を、上記除数レジスタ(1)の最下位ビットに置数する第2の手段と、

上記第1の手段と第2の手段の結果を上記減数レジスタ(3)で合成する第3の手段と、

上記演算結果であるキャリ信号の極性に応じて、上記部分剰余レジスタ(5)の前サイクルの結果( $R_{n-1}$ )を選択するか、今回の演算によって得られた部分剰余( $R_n$ )を選択する切り替え手段(6)とを設けて、

上記基数2のオペランドから開平法によって平方根を求める演算式、

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

には $R_{n-1}$ を、それ以外の場合は $R_n$ を2ビット左シフトしたものを部分剰余とし、

前サイクル迄の中間結果を1ビット左シフトした結果に、上記部分解を加えることを繰り返して平方根を求める平方根演算処理方式に関し、

開平法による平方根演算を、引き離し法による高基数非回復型除算装置を用いて1演算サイクルで1桁の平方根を得ることを目的とし、

少なくとも、除数レジスタと、除倍数回路と、該除倍数の結果を保持する減数レジスタとを備えた高基数非回復型除算装置に、オペランドを置数する被開平方レジスタと、上記置数されたオペランドを1演算サイクル毎に2ビット左シフトして、上記高基数非回復型除算装置の部分剰余レジスタに伝達する手段と、上記演算サイクル毎に、上記除数レジスタを1ビット左シフトして帰還する手段と、上記部分剰余レジスタの値から上記除数レジスタの値を1ビット左シフトして4倍したものを減算したときに、加算器から得られるキャリ信号を、上記除数レジスタの最下位ビットに帰還し

て、平方根の部分解を得る手段と、上記演算サイクル毎に、部分剰余レジスタを2ビット左シフトしたものと、上記減算処理によって得られる部分剰余( $R_n$ )との何れかを、上記キャリ信号によって選択して、上記部分剰余レジスタに置数して、当該演算サイクルでの部分剰余とする手段とを設けて、当該高基数非回復型除算装置を用いて平方根演算処理を行う方式において、前サイクルでの部分剰余を置数した部分剰余レジスタから、減算すべき前サイクルの部分中間結果である( $A_{n-1}$ )の $(A_{n-1}) \times 4$ を生成するのに、1サイクル前の中間結果を1ビット左シフトして'×4'を行う第1の手段と、今回の演算サイクルで得られた部分解(キャリ)を1ビット左シフトして'×4'を行うと同時に、該部分解(キャリ)を、上記除数レジスタの最下位ビットに置数する第2の手段と、上記第1の手段と第2の手段の結果を上記減数レジスタで合成する第3の手段と、上記演算結果であるキャリ信号の極性に応じて、上記部分剰余レジスタの前サイクルの結果( $R_{n-1}$ )を選択するか、今

### 3. 発明の詳細な説明

#### (目次)

##### 概要

##### 産業上の利用分野

##### 従来の技術と発明が解決しようとする問題点

##### 問題点を解決するための手段

##### 作用

##### 実施例

##### 発明の効果

#### (概要)

基数2のオペランドから開平法によって平方根を求める演算式

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

但し、 $R_{n-1}$  : 前サイクルでの余り

$A_{n-1}$  : 前サイクル迄の中間結果

を計算し、 $R_n < 0$ であれば'0'を、それ以外であれば'1'を部分解とし、且つ、 $R_n < 0$ の場合

回の演算によって得られた部分剰余( $R_n$ )を選択する切り替え手段とを設けて、上記基数2のオペランドから開平方によって平方根を求める演算式、

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

を処理するように構成する。

#### (産業上の利用分野)

本発明は、基数2のオペランドから開平方によって平方根を求める演算式

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

但し、 $R_{n-1}$  : 前サイクルでの余り

$A_{n-1}$  : 前サイクル迄の中間結果

を計算し、 $R_n < 0$  であれば '0' を、それ以外であれば '1' を部分解とし、且つ、 $R_n < 0$  の場合には  $R_{n-1}$  を、それ以外の場合は  $R_n$  を2ビット左シフトしたものを部分剰余とし、

上記部分解を、前サイクル迄の中間結果を1ビット左シフトした結果に加えることを繰り返して平方根を求める平方根演算処理方式に関する。

最近の計算機技術の進歩に伴って、該計算機に

よるデータ処理の多様化と、処理量の増加が進展し、該計算機に対する処理能力の向上が求められており、科学技術計算の分野で用いられる平方根演算についても、その高速化が要求されるが、該平方根の演算を高速に処理する為に、専用の平方根演算回路を設けることはハードウェア量の増加が大きいことから、使用頻度の比較的小さい平方根演算を経済的に、且つ高速に処理できる平方根演算処理方式が求められることになる。

一方、平方根演算は、高基数非回復型除算装置と云った、所謂引き離し法の除算方式とその手順が類似していることから、例えば、上記高基数非回復型除算装置に、該平方根演算に必要な回路を、該除算装置での除算速度に影響を与えない方法で付加して行う平方根演算処理方式が知られている。

上記引き離し法の除算方式によって高速の除算を実現することを目的として、除倍数を減数レジスタに置数する形式の高基数非回復型除算装置の構成法を、本願出願者は特開昭 60-160438号公報で開示しているが、このような高基数非回復型除

算装置を使用して平方根演算を行う場合、平方根の中間結果、即ち、部分解が除数レジスタ(DSR)の最下位ビットに帰還される為、1桁の平方根を得るのに、演算結果のキャリ信号を除数レジスタ(DSR)にセットするサイクルと、該除数レジスタ(DSR)の値に基づいて、除倍数×除数(具体的には、除数の4倍数)を求めるサイクルの2演算サイクルを必要とする問題があり、この処理を該除算方式に影響を与えることなく1演算サイクルで行う平方根演算処理方式が平方根演算の高速化に必要となる。

#### (従来の技術と発明が解決しようとする問題点)

第4図は従来の平方根演算処理方式を説明する図であって、(a)は除算回路を示し、(b)は該除算回路を用いて、1演算サイクルで平方根の1桁をえる平方根演算回路を示している。

一般に、平方根を求める手順において、各演算サイクルでのオペランドと中間結果(即ち、部分解)との間には、

$$(ar)^2 \leq \text{オペランド} = (ar + b)^2 + c \\ < ((a+1)r)^2 = (ar + r)^2 \quad \text{..... ①}$$

と表すことができる関係がある。

但し、 $ar$  = 前サイクル迄に求めた中間結果

$r$  = 基数

$a, b$  = 整数

$c$  = 実数  $\leq r$

である。

これより、 $c$  を最小( $\geq 0$ )にするように  $b$  を計算すれば次の中間結果を求めることができる。

ここで、オペランドから中間結果の2乗を引いた余りを " $Rc$ " とすると、

$$Rc = \text{オペランド} - (ar)^2 \quad \text{..... ②}$$

と書けるので、①式と合わせて変形すると、

$$Rc = (ar + b)^2 + c - (ar)^2 \\ = (ar)^2 + 2arb + b^2 + c - (ar)^2 \\ = 2arb + b^2 + c$$

$$\therefore Rc - (2arb + b^2) = c \quad \text{..... ③}$$

一方、 $c$  は  $r \geq c \geq 0$  であるから、 $b$  の範囲

は上記④式より、

$$0 \leq b \leq r-1 \text{-----} \textcircled{4}$$

となる。

ここで、 $r=2$ 、即ち、2進数の場合の平方根の演算処理方式について考えてみると、④式より、

$$b=0, \text{又は} 1 \text{となる。}$$

従って、 $b=1$ と仮定して、上記④式に代入すると、

$$Rc = (2 \times a \times 2 \times 1 + 1^2) = c$$

$$\therefore c = Rc = (4 \cdot a + 1) \text{-----} \textcircled{5}$$

これより、

$$Rc = (4 \cdot a + 1) \geq 0 \text{のとき、} b=1$$

$$< 0 \text{のとき、} b=0$$

と該中間結果  $b$  を決めることができる。そして、 $c$  は次のサイクルの余り ' $Rc$ ' となる。

上記の⑤式は、引き離し法による除算の演算式である、

$$R_1 = r \cdot R_0 + m \cdot D$$

但し、 $D$  = 除数

$r$  = 基数

除倍数  $\times$  除数 ( $mD$ ) を置数する為の減数レジスタ ( $SR1, 2, 3$ ) を設けて、1 演算サイクルタイムを短くして高速の除算を行うことができる高基数非回復型除算装置を、前述の特開昭 60-160438号公報に開示している。

この方式においては、上記除倍数  $\times$  除数 ( $mD$ ) の全てのケースを予め計算して、減数レジスタ ( $SR1 \sim 3$ ) に置数できるようにしておき、各演算サイクルにおいて出力される部分商予測値  $m$  によって、その一つを選択し、該減数レジスタ ( $SR1 \sim 3$ ) に置数することで、1 演算サイクルタイムを短くして、且つ該 1 演算サイクルタイムで 1 桁の除算結果を得るものである。

然しながら、この高基数非回復型除算装置を用いて、上記開平法による平方根演算を行う場合には、該除算装置の加算回路のキャリとして得られる 1 桁の部分解を、当該除算回路の除数レジスタ ( $DSR$ ) の最下位ビットに帰還して更新する必要がある為、前述の、基数 2 のオペランドから開平法によって平方根を求める演算式

$m$  = 除倍数 (部分予測商)

と類似しており、該引き離し法による除算回路を用いて平方根演算ができることが分かる。

第 4 図の (a) は該除算回路の構成例を示したものであり、(b) は該除算回路を用いて、上記④式に基づいて平方根演算回路を構成した例を示したものである。

上記平方根演算方式の導出過程、及び構成例については、例えば、文献「除算演算と平方根演算の為の共用ハードウェア」、ジョージ S. テイラー著、米国電気電子工学協会、1981、("COMPATIBLE HARDWARE FOR DIVISION AND SQUARE ROOT" George S. Taylor, IEEE, 1981) に示されているが、本図からも明らかな如く、この除算方式においては、除倍数  $\times$  除数 ( $mD$ ) を置数する為の減数レジスタ ( $SR$ ) を持っていない為、除算での 1 演算サイクルに必要な時間が長くなると言う問題がある。

そこで、本発明者は、除数 ( $D$ ) は最初に一度除数レジスタ ( $DSR$ ) に置数すると、該除算の演算が終了する迄変化することがないことに着目して、

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

但し、 $R_{n-1}$  : 前サイクルでの余り ( $Rc$ ),

$A_{n-1}$  : 前サイクル迄の中間結果 ( $a$ )

を計算し、 $R_n < 0$  であれば '0' を、それ以外であれば '1' を部分解とし、且つ、 $R_n < 0$  の場合には  $R_{n-1}$  を、それ以外の場合は  $R_n$  を 2 ビット左シフトしたものを部分剰余とし、

上記部分解を、前サイクル迄の中間結果を 1 ビット左シフトした結果に加えることを繰り返して平方根を求める演算を行おうとすると、「 $4 \cdot A_{n-1}$ 」を減数レジスタ ( $SR1 \sim 3$ ) に置数するサイクルと、該置数された「 $4 \cdot A_{n-1}$ 」を用いて前サイクルの部分剰余「 $R_{n-1}$ 」から減算するサイクルの 2 サイクルを必要とし、当該高速化された高基数非回復型除算装置を用いて平方根演算を行う場合には高速化できないと言う問題があった。

本発明は上記従来の欠点に鑑み、除倍数  $\times$  除数 ( $mD$ ) の結果を、そのサイクルで得られる部分商予測値  $m$  に基づいて選択し、減数レジスタ ( $SR1 \sim 3$ ) に置数する方式の高基数非回復型除算装置を用い

て、基数2のオペランドから開平法によって平方根を求める演算式

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

但し、 $R_{n-1}$  : 前サイクルでの余り(Rc),

$A_{n-1}$  : 前サイクル迄の中間結果(a)

の演算を1演算サイクルで行うことで、1桁の平方根(部分解)を1演算サイクルで得る平方根演算処理方式を提供することを目的とするものである。

(問題点を解決するための手段)

第1図は本発明の平方根演算処理方式の原理図である。

上記の問題点は、下記の如くに構成された平方根演算処理方式によって解決される。

基数2のオペランドから開平法によって平方根を求める演算式

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

但し、 $R_{n-1}$  : 前サイクルでの余り

$A_{n-1}$  : 前サイクル迄の中間結果

得られるキャリ信号(CARRY)を、上記除数レジスタ(DSR)1の最下位ビットに帰還して、平方根の部分解を得る手段④と、

上記演算サイクル毎に、部分剰余レジスタ(PR)5を2ビット左シフトしたものと( $R_{n-1}$ )と、上記減算処理によって得られる部分剰余( $R_n$ )との何れかを、上記キャリ信号(CARRY)の値によって選択して、上記部分剰余レジスタ5に置数し、当該演算サイクルでの部分剰余とする手段⑤とを設けて、

当該高基数非回復型除算装置を用いて平方根演算処理を行う方式において、

上記オペランドの上位2ビットを置数した部分剰余レジスタ(PR)5から減算すべき前サイクルの部分中間結果( $A_{n-1}$ )の $(A_{n-1}) \times 4$ を生成するのに、

1サイクル前の中間結果( $A_{n-1}$ )(DSR)1を1ビット左シフトして'×4'を行う第1の手段と、

今回の演算サイクルで得られた部分解(キャリ)を1ビット左シフトして'×4'を行うと同時に、該部分解(キャリ)を、上記除数レジスタ(DSR)1の最下位ビットに置数する第2の手段と、

を計算し、 $R_n < 0$ であれば'0'を、それ以外であれば'1'を部分解とし、且つ、 $R_n < 0$ の場合には $R_{n-1}$ を、それ以外の場合は $R_n$ を2ビット左シフトしたものを部分剰余とし、

上記部分解を、前サイクル迄の中間結果を1ビット左シフトした結果に加えることを繰り返して平方根を求めるのに、

少なくとも、除数レジスタ1と、除倍数回路2と、該除倍数×除数の結果を保持する減数レジスタ3とを備えた高基数非回復型除算装置に、オペランドを置数する被開平方数レジスタ(SQR)6と、

上記置数されたオペランドを1演算サイクル毎に2ビット左シフトして、上記高基数非回復型除算装置の部分剰余レジスタ(PR)5に伝達する手段①と、

上記演算サイクル毎に、上記除数レジスタ(DSR)1を1ビット左シフトして帰還する手段②と、

該部分剰余レジスタ(PR)5の値から上記除数レジスタ(DSR)1の値を1ビット左シフトして4倍したものを減算したときに、加算器(ADD)4から

上記第1の手段と第2の手段の結果を上記減数レジスタ(SR3)3で合成する第3の手段③と、

上記演算結果であるキャリ信号(CARRY)の値に応じて、上記部分剰余レジスタ(PR)5の前サイクルの結果( $R_{n-1}$ )を選択するか、今回の演算によって得られた部分剰余( $R_n$ )を選択する切り替え手段⑥とを設けて、

上記基数2のオペランドから開平法によって平方根を求める演算式、

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

を処理するように構成する。

(作用)

即ち、本発明によれば、基数2のオペランドから開平法によって平方根を求める演算式

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

但し、 $R_{n-1}$  : 前サイクルでの余り

$A_{n-1}$  : 前サイクル迄の中間結果

を計算し、 $R_n < 0$ であれば'0'を、それ以外であれば'1'を部分解とし、且つ、 $R_n < 0$ の場合

には $R_{n-1}$ を、それ以外の場合は $R_n$ を2ビット左シフトしたものを部分剰余とし、

上記部分剰余を、前サイクル迄の中間結果を1ビット左シフトした結果に加えることを繰り返して平方根を求める方式である。

上記の演算式における減算を補数を用いた加算に書き替えると、

$$\begin{aligned} R_n &= R_{n-1} - (4 \cdot A_{n-1} + 1) \\ &= R_{n-1} - 4 \cdot A_{n-1} - 1 \\ &= R_{n-1} + \overline{4 \cdot A_{n-1} + 1} - 1 \\ &= R_{n-1} + \overline{4 \cdot A_{n-1}} \quad \text{..... ⑦} \end{aligned}$$

となる。

この加算を行うのに必要な $\overline{4 \cdot A_{n-1}}$ を求める場合、従来方式においては、除数レジスタを1ビット左シフトすると同時に、該平方根演算で求めた部分剰余であるキャリ信号(CARRY)を、1ビット左シフトした上記除数レジスタ(DSR)の最下位ビットに帰還して更新し、該更新した除数レジスタ(DSR)の1の補数をとったものを4倍して、減数レジスタ(SR3)にセットする処理となる為、1桁の

部分剰余を得るのに2演算サイクルを必要としていた。

そこで、本発明においては、減数レジスタ(SR3)にセットする値を上記更新した除数レジスタ(DSR)からではなく、更新前、即ち、前サイクルの除数レジスタ(DSR)の値を1ビット左シフト(即ち、2倍)したものの4倍したもの(従って、8倍したもの)の1の補数をとったものと、当該演算サイクルで得られた部分剰余(キャリ信号(CARRY))を8倍、即ち、3ビット左シフトしたものの1の補数をとったものとを、該減数レジスタにセットするようにする。

第3図は本発明による $\overline{4 \cdot A_{n-1}}$ の演算方式を説明する図であって、(a)は除数レジスタ(DSR)にキャリ信号(CARRY)を帰還した後において、1の補数をとる場合を示しており、(b)は上記の本発明による $\overline{4 \cdot A_{n-1}}$ の処理方式(即ち、1演算サイクルで求める方式)を示しており、両者とも同じ結果が得られることが分かる。

従って、本発明においては、減数レジスタを備

えた引き離し法による除算装置を用いて平方根演算を行う際に必要な、

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

なる演算を1演算サイクルで実行でき、最小限のハードウェアを付加するだけで、高速の平方根演算処理方式が実現できる効果がある。

#### 〔実施例〕

以下本発明の実施例を図面によって詳述する。

前述の第1図が本発明の平方根演算処理方式の原理図であり、第2図は本発明の一実施例を模式的に示した図であって、第1図、第2図における被開平レジスタ(SQR)6、除数レジスタ(DSR)1を1ビット左シフト(1 bit-LEFT-SHIFT)する手段③、部分剰余レジスタ(PR)5を2ビット左シフト(2 bit-LEFT-SHIFT)する手段②、加算器(ADD)4からのキャリ信号(CARRY)と、前サイクルの除数レジスタ(DSR)とを用いて、 $\overline{4 \cdot A_{n-1}}$ を減数レジスタ(SR3)3で合成する手段④、及び該キャリ信号(CARRY)の値によって、部分剰余レジスタ(PR)5に

セットする部分剰余を選択する手段⑤が本発明を実施するのに必要な手段である。尚、全図を通して、同じ符号は同じ対象物を示している。

以下、第1図～第3図を用いて、本発明による平方根演算処理方式を説明する。

本発明を実施しても、引き離し法による除算装置を用いて平方根を求める基本的な動作は従来方式と特に変わることはないので省略し、ここでは、前述の基数2の開平演算式、

$$\begin{aligned} R_n &= R_{n-1} - (4 \cdot A_{n-1} + 1) \\ &= R_{n-1} - 4 \cdot A_{n-1} - 1 \\ &= R_{n-1} + \overline{4 \cdot A_{n-1} + 1} - 1 \\ &= R_{n-1} + \overline{4 \cdot A_{n-1}} \quad \text{..... ⑦} \end{aligned}$$

で必要となる $\overline{4 \cdot A_{n-1}}$ を1演算サイクルで求める処理を中心にして、第2図を用いて説明する。

先ず、第2図に示したH0サイクルにおいて、オペランドが被開平レジスタ(SQR)6にセットされると共に、除数レジスタ(DSR)1、減数レジスタ(SR1～3)3、部分剰余レジスタ(PR)5が、それぞれクリア('0'セット)される。

続いて、同じサイクルにおいて、被開平レジスタ(SQR) 6 の上位2ビットが最初の平方根を求めるのに、部分剰余レジスタ(PR) 5の最下位2ビットに伝達①されると共に、該被開平レジスタ(SQR) 6 の最下位2ビットに '00' が挿入される。

又、除数レジスタ(DSR) 1 の1の補数をとったものを3ビットシフト、即ち、上記開平演算式に必要な $4 \times A_{n-1}$ を求める為の4倍と、1演算処理が行われる毎に、得られた平方根(キャリ)が該除数レジスタ(DSR) 1 の最下位1ビットに置数する為に、該除数レジスタ(DSR) 1 を1ビット左シフト、即ち、2倍との合計8倍する必要がある為に、3ビット左シフト(これを、本図においては、 $DSR \times 8$ で示している)して減数レジスタ(SR3) 3 にセットする。

以下、1演算サイクルでの動作を詳細に説明する。

(1) 上記の式の $R_{n-1} + \overline{4 \times A_{n-1}}$ を行う為に、減数レジスタ(SR3) 3 と、部分剰余レジスタ(PR) 5 が加算器(ADD) 4 において加算される。

ルにおける1ビットシフトと、その結果を更に、4倍するための2ビットシフトを合わせたものである。

(4) 部分剰余レジスタ(PR) 5は演算結果であるキャリ(CARRY)の値によって、次の2通りのセットが行われる。

該キャリ(CARRY)が'1'であれば、加算器(ADD) 4の結果が2ビット左シフト①され(12サイクル参照)、該シフト後の下位2ビットに、被開平数レジスタ(SQR) 6の上位2ビットがシフトイン①される。

該キャリ(CARRY)が'0'であれば、前サイクルの部分剰余レジスタ(PR) 5が2ビット左シフト②されて、該シフト後の下位2ビットに、被開平数レジスタ(SQR) 6の上位2ビットがシフトイン①される。

以下、同じ動作が11～の各サイクル毎に、除数レジスタ(DSR) 1に、上記加算器(ADD) 4のキャリ(CARRY)信号がシフトイン(CARRY SHIFT-IN Cで示す)されて保持されていく中間結果が必要

(2) 除数レジスタ(DSR) 1は左に1ビットシフト③され、その最下位ビットに上記の加算結果のキャリ(CARRY)がシフトイン(CARRY SHIFT-IN C)される。

ここでは、前述のように、補数の加算によって減算を行っているので、該減算の結果が正の場合には、キャリ(CARRY)が'1'となるため、該キャリ(CARRY)の値がその偽部分解となる。

従って、減数レジスタ(SR3) 3にセットする場合には、その1の補数をとった'c'を入力する(CARRY SHIFT-IN  $\overline{C}$ )必要がある。

(3) 減数レジスタ(SR3) 3には、前サイクルの除数レジスタ(DSR) 1の1の補数を左に3ビットシフトしたもの( $\overline{DSR \times 8}$ )をセットし、最下位から3ビット目に、上記キャリの1の補数'c'をセットし、下位2ビットには、'00'の1の補数である'11'をセット(SHIFT-IN)する。(第9図(b)参照)

ここで、除数レジスタ(DSR) 1の1の補数を3ビットシフトするのは、前述のように、現サイク

な桁数になる迄繰り返される。

このように、本発明による平方根演算処理方式を用いれば、減数レジスタ(SR1~3)3を備えた高速度の除算装置を共用した開平演算回路で、1サイクル毎に1ビットの部分解を求めることができる。

このように、本発明は、引き離し法による、例えば、高基数非回復型除算装置での除算方式と、平方根の演算方式とが類似していることに着目し、該除算回路に最小限のハードウェアを付加して、平方根演算を行う方式において、該除算回路が除倍数 $\times$ 除数(mD)を減数レジスタ(SR1~3)に置数する形式の場合、除数レジスタ(DSR)の開平結果である加算器(ADD)のキャリ信号(CARRY)を帰還した後、該減数レジスタ(SR1~3)への置数処理が通常2演算サイクル必要となる処理を、1演算サイクルでできるように、前サイクルでの除数レジスタ(DSR)の1の補数をとって、3ビット左シフトしたものと、該演算結果であるキャリ信号(CARRY)の1の補数をとって3ビット左シフトしたも

のとを減数レジスタ(SR3)で合成すると共に、次の演算の為に該キャリ信号(CARRY)を除数レジスタ(DSR)の最下位ビットに帰還するようにした所に特徴がある。

#### 〔発明の効果〕

以上、詳細に説明したように、本発明の平方根演算処理方式は、基数2のオペランドから開平方によって平方根を求める演算式

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

但し、 $R_{n-1}$  : 前サイクルでの余り

$A_{n-1}$  : 前サイクル迄の中間結果

を計算し、 $R_n < 0$  であれば '0' を、それ以外であれば '1' を部分解とし、且つ、 $R_n < 0$  の場合には  $R_{n-1}$  を、それ以外の場合は  $R_n$  を2ビット左シフトしたものを部分剰余とし、

上記部分解を、前サイクル迄の中間結果を1ビット左シフトした結果に加えることを繰り返して平方根を求める方式において、

上記減算を補数を用いた加算に書き替えると、

$$-R_{n-1} + 4 \cdot A_{n-1} \quad \text{①}$$

となることから、この加算を行うのに必要な、

$$4 \cdot A_{n-1}$$

を求めるのに、減数レジスタ(SR3)にセットする値を、演算結果によって更新した除数レジスタ(DSR)からではなく、更新前、即ち、前サイクルの除数レジスタ(DSR)の値を1ビット左シフト(即ち、2倍)したものの4倍したものの1の補数をとったものと、当該演算サイクルで得られた部分解(キャリ信号(CARRY))を8倍、即ち、3ビット左シフトしたものの1の補数をとったものとを、該減数レジスタにセットするようにしたものである。このため、減数レジスタを備えた引き離し法による除算装置を用いて平方根演算を行う際に必要な、

$$R_n = R_{n-1} - (4 \cdot A_{n-1} + 1)$$

なる演算を1演算サイクルで実行でき、最小限のハードウェアを付加するだけで、高速の平方根演算処理方式が実現できる効果がある。

#### 4. 図面の簡単な説明

第1図は本発明による平方根演算処理方式の原理図。

第2図は本発明の一実施例を模式的に示した図。

第3図は本発明による  $4 \cdot A_{n-1}$  の演算方式を説明する図。

第4図は従来の平方根演算処理方式を説明する図である。

図面において、

1 は除数レジスタ(DSR)、2 は除倍数回路、

3 は減数レジスタ(SR1~3)、

4 は加算器(ADD)、

5 は部分剰余レジスタ(PB)、

6 は被開平方レジスタ(SQR)、

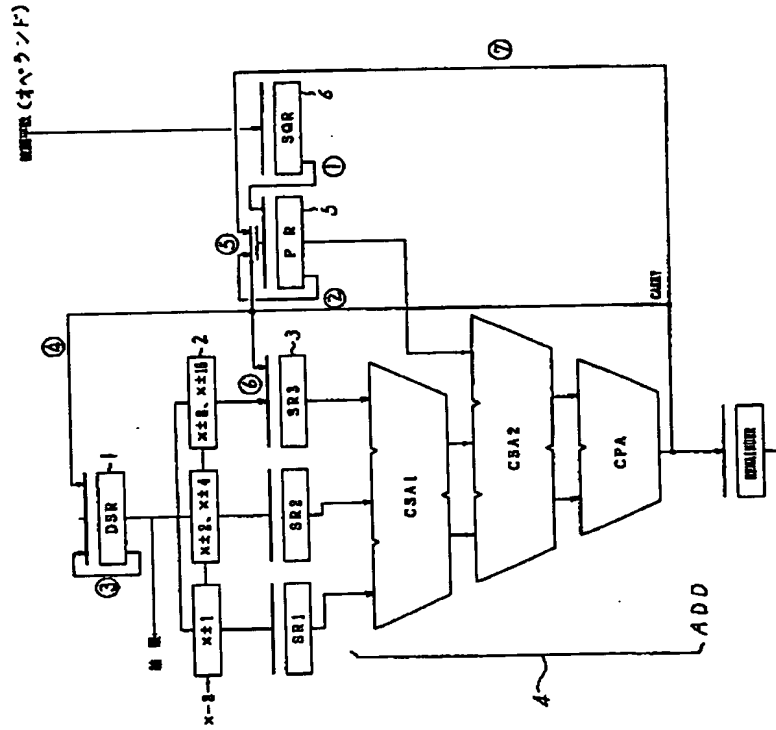
①~④はシフト等の動作、

CARRY はキャリ信号(C)、又は部分解、

をそれぞれ示す。

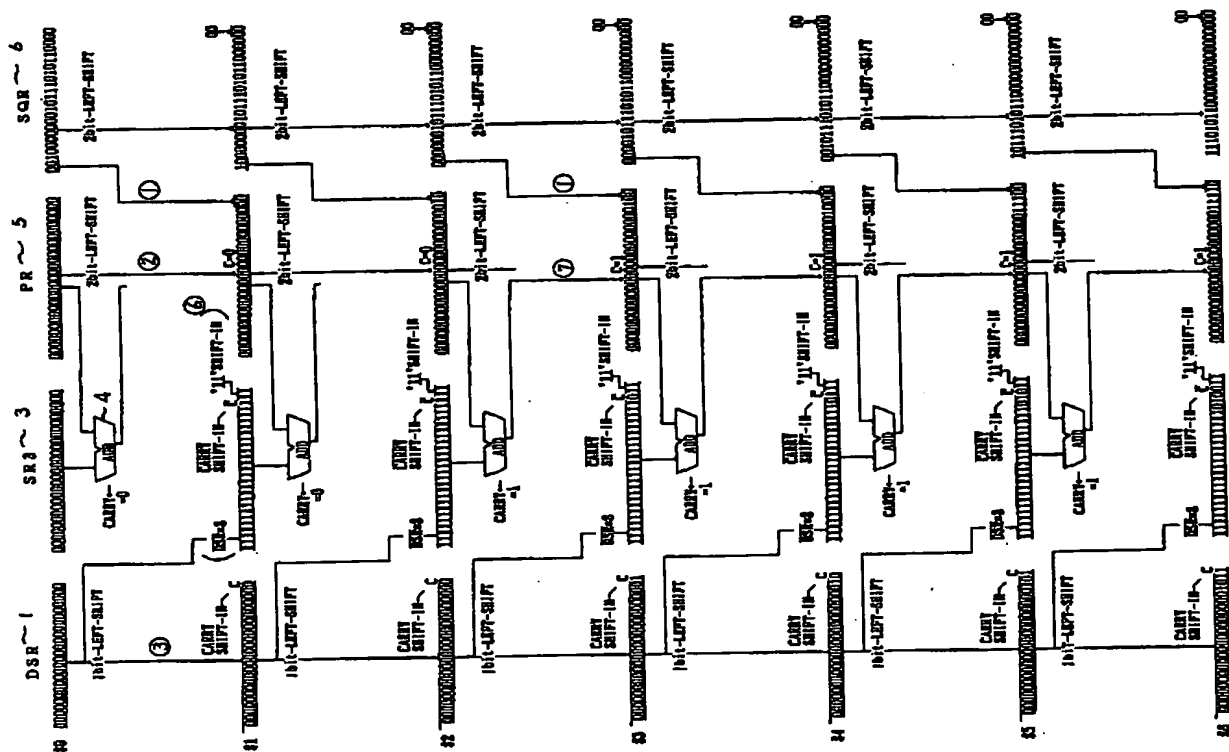
代理人 弁理士 井桁貞一





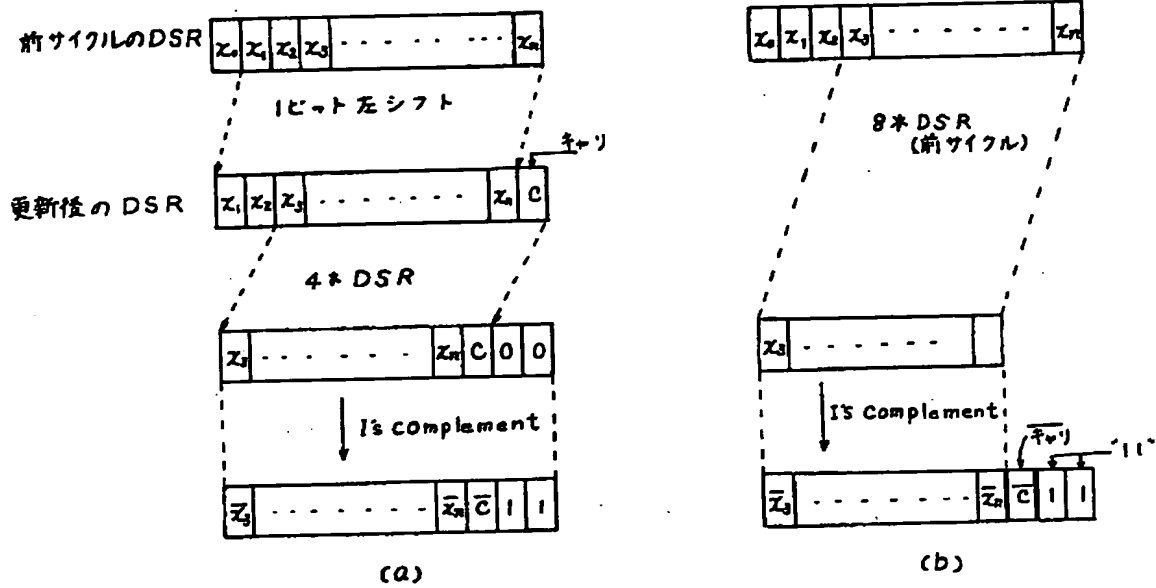
本発明平方根演算処理方式の原理図

第 1 図

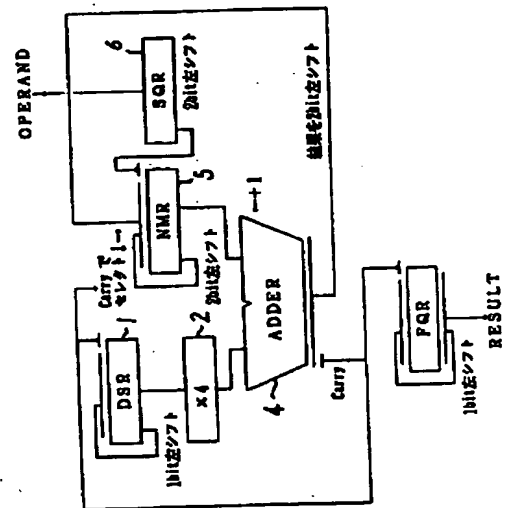
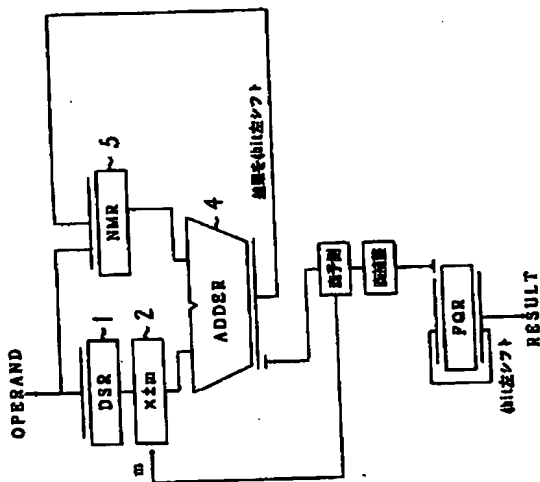


本発明の一実施例を模式的に示した図

第 2 図



本発明による  $4 * A_{n-1}$  の演算方式を説明する図  
第 3 図



従来の平方根演算処理方式を説明する図  
第 4 図